

## SYSTEM FOR MANAGEMENT OF TRANSACTIONS ON NETWORKS

### COPYRIGHT REFERENCE

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

### CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims the benefit under 35 U.S.C. §119(e) of U.S. Provisional Application No. 60/040,223 filed March 7, 1997, <sup>both of which</sup> ~~which~~ is hereby incorporated by reference.

### STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

This invention has been created without the sponsorship or funding of any federally sponsored research or development program.

### REFERENCE TO MICROFICHE APPENDIX

This application includes a microfiche appendix in accordance with 37 C.F.R. 1.96(c) comprising ten (10) microfiche having a total of 600 frames. The appendix is hereby incorporated by reference.

The appendix contains a source tree for programs embodying the present invention including: the National Center for Supercomputing Applications httpd 1.4.2 web server with customizations in accordance with the present invention, the "cs-log" and "cs-logd" logging daemons in accordance with the present invention; and the "tvsd" token-management daemon according to the present invention.

## BACKGROUND OF THE INVENTION

The open-systems Internet offers a very different environment to users when compared with traditional consumer on-line services and Electronic Data Interchange (EDI) value-added networks (VANs). Specifically, the Internet environment is very decentralized, and no one organization controls the user base or access to resources. While this decentralization has tremendous advantages (chief among them, the freedom to select from a wide number of service and content offerings), this lack of "unity" can confuse and sometimes frustrate both potential information providers and users. In addition, the "stateless protocol" of the Internet presents difficulties for entities which need to know the identity and usage requirements of their service clients.

A challenge is to provide a more unified user-access environment without destroying the freedom of choice inherent in the Internet model. One aspect of this challenge is that the unlimited variety of kinds of transactions which occur on the Internet include some kinds of transactions which are very difficult to manage cost-effectively. One such kind is the low-priced transaction.

The prior art systems and methods for managing transactions on networks have been too complex and too expensive to use effectively on low-priced transactions. These and other difficulties experienced with the prior art systems and methods have been obviated in a novel manner by the present invention.

It is, therefore, an outstanding object of the present invention to provide a system and method for managing transactions on networks.

Another object of this invention is to provide a system and method for managing transactions on networks which is so simple and so inexpensive to use that it can be effectively used to manage low-priced transactions.

A further object of the present invention is to provide a system for managing client accounts and controlling access to resources over data networks.

It is another object of the invention is to provide a system which provides a mechanism for sharing client information and charges among a plurality of service providers.

5 A still further object of the invention is to provide a system which includes a client who is registered with one of the service providers (the "home provider") and is allowed to access the resources of the other service providers ("outside providers") that are part of the system.

It is a further object of the invention to provide a system which includes a settling means adapted to allow the system to settle accounts among service providers by charging the home provider for access by its clients to the resources of the outside providers.

10 It is a further object of the invention to provide a system which includes a payment means adapted to assure that the outside providers are then paid for access through the system.

15 It is a further object of the invention to provide a system which includes a sharing means adapted to allow the system to allow the providers to share users without requiring an open account for each user at each provider.

It is a further object of the invention to provide a system which includes a verification means adapted to allow each provider to determine if a particular client is a member of the system, verify that the client has authenticated at his home provider, and determine this client's access privileges and criteria.

20 20 It is a further object of the present invention to provide a system for identifying client users for a variety of purposes including personalization and audience measurement.

25 It is a further object of the invention to provide a system and method for managing transactions on networks which is capable of being developed of high quality and at a low cost, and which is capable of providing a long and useful life with a minimum of maintenance.

5

With these and other objects in view, as will be apparent to those skilled in the art, the invention resides in the combination of parts and steps set forth in the specification and covered by the claims appended hereto, it being understood that changes in the precise embodiment of the invention herein disclosed may be made within the scope of what is claimed without departing from the spirit of the invention.

#### BRIEF SUMMARY OF THE INVENTION

10 1.0. This document describes the Token Validation Service (TVS) user and access management environment. Clickshare identifies the operator of the system. TVS is:

15 1.1. A system for managing client accounts and controlling access to resources over data networks which includes a mechanism for sharing client information and charges among a plurality of service providers. A client who is registered with one of the service providers (the "home provider") is allowed to access the resources of the other service providers ("outside providers") that are part of the system. The system settles accounts among service providers by charging the home provider for access by its clients to the resources of the outside providers. The outside providers are then paid for that access through the system. Thus, the system allows the providers to share users without requiring an open account for each user at each provider. The system includes a mechanism by which each provider can determine if a particular client is a member of the system, verify that the client has authenticated at his home provider, and determine this client's access privileges and criteria.

20

TVS is a service for validating and profiling a large base of users distributed across independent content providers, simultaneously supporting content usage verification ("audience measurement") and billing at the "micro-transaction" ("per-page") level.

25

TVS is designed for use in an open-systems, public-network environment such as the Internet. TVS is implemented using an object-oriented client/server approach to enhance both scalability (in terms of number of users and number of transactions) and portability among operating environments.

TVS servers are operated by any Clickshare-authorized licensee of the server-side software. The "clients" of this service are, currently, HyperText Transfer Protocol (HTTP) servers operated by licensed, but otherwise independent, content providers. HTTP is the supporting protocol for the World Wide Web, the Internet's popular user environment.

5 Using TVS, content providers can "share users" through a common validation/profiling technique and exchange value for their content through a common, background, settlement process. By permitting owners of content to collect royalties and receive commissions automatically, TVS creates the economic incentive for content providers to link to each other's content in a manner that leverages the content base of all providers simultaneously, and is completely transparent to the user.

10

A "TVS enabled" HTTP reference server running on the Unix(tm) operating system was developed.

Clickshare provides a "reference implementation" of the TVS client based on the 1.4.2 release of the HTTP server from the National Center for Supercomputing Applications (NCSA) - a well-understood public domain HTTP server that runs on many versions of the Unix(tm) operating system (including "free" Linux). This "TVS-enabled" HTTP server is written in the C programming language. The TVS service is available through a C language API with a clean and narrow interface to the original HTTP server code. The API is a fully separate library and can be used to build other implementations.

20 **BRIEF DESCRIPTION OF THE DRAWINGS**

2.0. The character of the invention, however, may best be understood by reference to one of its structural forms, as illustrated by the accompanying drawings, in which:

25 Figure 1 is a diagrammatic representation of a simple version of a system for management of transactions over a network, which system embodies the principles of the present invention, and

Figure 2 is a diagrammatic representation of a more complex version of a system for management of transactions over a network, which system embodies the principles of the present invention.

**DETAILED DESCRIPTION OF THE INVENTION**

**5 3.1 COMMERCIAL PURPOSE OF INVENTION**

The Clickshare/TVS Service is a distributed user-management service for Internet information micropayments, access control, audience measurement and personalization with one-ID, one-bill user convenience. It is designed to address the problem of how to charge Internet users for their use of resources and control their access to those resources. It is also designed to provide for the transfer of information about users among multiple web sites in order to control access or define service authorization. TVS enables:

**MICROPAYMENTS** -- It makes it possible for publishers, and information or software owners to sell easily to Internet consumers in units as little as 10 cents per item -- so called "micropayments".

**15 PERSONALIZATION** -- It allows consumers to store their custom information preferences as part of their user profile and then optionally give those preferences to web publishers who wish to personalize their offerings. Clickshare's "Digital Calling Card" technology makes this possible.

**20 ACCESS CONTROL** -- It permits a web site to differentiate requests for information by individual users rather than broad domains -- even if the user has never registered with that particular web site. This "Service Class" technology avoids users having to maintain multiple IDs and passwords.

**25 AUDIENCE MEASUREMENT** -- Advertisers want to measure the effectiveness of their pitches by knowing as much about individual viewers as possible. Basic Internet protocols identify users only by "domain." Clickshare's "Digital Calling Card" technology transfers

a unique identifier for each user worldwide. This creates a platform for fine-grained demographic analysis while protecting user privacy.

entity -- to purchase information at multiple web sites with single-ID and password convenience. No end-user software is required beyond a standard Web browser.

### 3.2. PARTIES INVOLVED IN SERVICE

The parties involved in the Clickshare/TVS Service include:

**INFORMATION SELLERS/RESOURCE PROVIDERS** -- Operators of World Wide Web sites who wish to make money from the sale of information or software, or wish to control access to resources. These are called Clickshare Publishing Members or Clickshare Resource Providers. Examples include: newspapers, magazines, specialty publications, new-media entrepreneurs, game vendors, software publishers, health-care providers, network or other service providers.

**BILLING AGENTS/SERVICE PROVIDERS** -- Consumers have preexisting, ongoing credit relationships with billing agents or service providers who agree to become Clickshare Service Providers. In exchange for a negotiated share of the "Clickstream" revenue from information sales, or for other consideration, these service providers assume responsibility for servicing and billing consumer or enterprise end users and for authenticating the user at the start of a Clickshare/TVS session. Examples include: Internet Service Providers, newspapers, specialized publishers, online services, telephone companies, cable and utility companies, credit-card issuing banks, health-care providers, retailers, other consumer-credit entities, network or other service providers and other enterprises.

... Examples include: Internet Service Providers, newspapers, specialized publishers, online services, telephone companies, cable and utility companies, credit-card issuing banks, health-care providers, retailers, other consumer-credit entities, network or other service providers and other enterprises.

Clickshare Users. They are customers of their billing agent and have no direct relationship with Clickshare Corp.

CLICKSHARE SERVICE CORP. -- Facilitating the authentication of Clickshare Users, and storing records of their access to web sites is the Clickshare Access and Logging Service (CALS). Operated by Clickshare Service Corp. or its licensees, CALS is a fault-tolerant network of one or many Internet servers which exchange real-time, encoded information with machines operated by information sellers and billing agents.

### **3.4 VALUE OF SERVICE TO PARTICIPANTS**

#### **INFORMATION SELLERS**

- 10 A way to get paid for selling information on the Internet
- 15 A third revenue stream after advertising and subscriptions
- A digital equivalent of "single-copy sales"
- Can provide user demographic and preference information with user consent
- Enhances customer service through ability to personalize
- Enables site access by service class such as subscriber only
- Produces auditable, third-party "page-view" data for advertisers
- Low entry barrier, pay-as-you-profit cost structure

#### **BILLING AGENTS**

- 20 A new revenue stream -- selling information instead of access or goods
- Low entry barrier, pay-as-you-profit cost structure
- Leverage existing billing facility for profits at little incremental cost
- Provides credibility and co-marketing strength of an affiliate relationship
- Source of data on where customers are going for information
- Solidifies billing agent as "home page" for customer

**CUSTOMER**

Convenience of single ID and password for information access anywhere on Web

Single billing relationship for buying anywhere on the Web

Choice of billing agents

5 Ease of payment through single, periodic bill via existing credit facilities

Requires no special end-user software and no new end-user credit relationship

Instant point-and-click purchasing with authentication in background

No transfer of credit-card information across the Internet

10 Total control of who can use personal information

User's address optionally protected from unwanted mail

Parental control built and regulated by publisher not by government

Option for nightly tally of information purchases

**CLICKSHARE SERVICE CORP.**

Front-loaded revenues from CPM and CSP member enrollment fees

Cost-based revenues from per-enabled-user fees

Scaled, annuity revenue from per-click transaction fees

Service fees for audience measurement data, installation and support

Commissions on advertising sales (Adshare -- pay-per-view ads)

**3.3. OBJECTIVE SYSTEM BENEFITS**

20 **UNIVERSAL SETTLEMENT** -- Publishers and on-line services have begun exploring ways to compensate each other for the services they provide to users [Advertising Age, Jan. 20, 1997, "Pay per view: Web sites seek deals with ISPs"]. Such contractual relationships will rapidly become unmanageable because of the variety of sources of information and users and the need to have bilateral agreements among players. A single settlement facility, as with the long-distance telephone industry, is needed. Also, if users are forced to join information cartels of large publishers or user-owners, they will be denied choice and will be forced to accept bundled pricing.

5           UNIVERSAL CREDIT -- Many web sites are enrolling users and accepting credit-card payments. But each of these relationships works only for that web site, much as a store-credit card issued by Sears, does not work at Target or Pennies. The experience in the consumer credit industry of a gradual conversion of most such accounts to VISA or MasterCard-backed systems demonstrates the desire for consumers and marketers to have less credit facilities rather than more which are universal in their application.

10           UNIVERSAL ACCESS -- Publishers who seek to charge users on a subscription basis by definition exclude the vast majority of potential users who would buy a portion of the web site's offerings on a "per-click" basis. Clickshare, uniquely, offers the opportunity to "have it both ways." Just as conventional newspaper and magazine publishers have subscribers and single-copy sales, the Clickshare publisher can have subscribers, but also vend information to visiting Clickshare users "by the click." In the proprietary online world (West, Lexis-Nexis, Compuserve, Knight-Ridder-Dialog) this has not been technically feasible because of the lack of a universal public network, such as the Internet, that takes care of site access. Clickshare provides the vicarious billing relationship.

#### 4. OBJECTIVES OF THE INVENTION

##### 4.1. SUMMARY

20           The Clickshare/TVS Service is designed to enable a free marketplace for digital information in which business relationships among multiple service providers are reduced to a set of technical protocols providing for significant flexibility in the pricing of resources, the transfer of user attributes and the control of access to system resources, all on a real-time basis across open networks such as the Internet.

##### 4.2 OVERVIEW

25           The open-systems Internet offers a very different environment to users when compared with traditional consumer on-line services and Electronic Data Interchange (EDI) value-added networks (VANs). Specifically, the Internet environment is very decentralized, and no one

organization controls the user base or access to resources. While this decentralization has tremendous advantages (chief among them, the freedom to select from a wide number of service and content offerings), This lack of "unity" can confuse and sometimes frustrate both potential information providers and users. In addition, the "stateless protocol" of the Internet presents difficulties for entities which need to know the identity and usage requirements of their service clients.

A challenge is to provide a more unified user-access environment without destroying the freedom of choice inherent in the Internet model. Clickshare's effort in this arena is called the Clickshare/TVS Service -- a network service that allows independent content and service providers to cooperate on user registration, user profiling and verification of site usage in a manner that allows:

- (a) Cross-site usage to be billed at the micro-transaction (per-page) level (that is, "settled") if/when desired;
- (b) Delivery of resources and the control of access according to the verified and submitted characteristics of the requesting user;
- (c) Transfer of data about user characteristics, including levels of authorization and service classes, among multiple service providers.
- (d) Central logging of access by users to system resources among multiple service providers for billing or other user-management or demographic purposes.
- (e) Transfer of pricing relationships or data among original publishers of information, including software, and their billing or sales agents or brokers.

TVS does not require changes to the user "client" software (i.e. web browsers), and does not negatively impact any of the several credit-card and e-cash transaction facilities being deployed or discussed. Rather, it is designed to enable a class of transactions in a value range (5 cents and up) that is uneconomic as single credit-card purchases -- and bundle

them for monthly or weekly settlement. Thus, TVS is compatible with most announced e-cash implementations.

#### 4.3 KEY DESIGN FEATURES

##### 4.3.1. CLIENT INDEPENDENT, TRANSPARENT TO USERS

5 One method of unifying the Internet is to create a single application or application suite that accesses all services, then attempt to get all users to use it exclusively, possibly using proprietary features that users want (but other vendors can't copy). Instead, the TVS approach is to have low impact on client software, and where impact is required, adherence to Internet standards. TVS is transparent to users in that it does not introduce impediments into usage of Internet services (specifically, in its first offering, the World Wide Web), and has minimal performance impact.

##### 4.3.2. SCALABLE SERVICE, CAPABLE OF HIGH VOLUME OF USERS AND USAGE TRAFFIC

10 Internet usage is predicted to grow dramatically as high-quality and exciting services are made available on the Internet. TVS is designed to scale as the number of content providers, users, and exciting services are made available on the Internet. TVS is designed to scale as the number of content providers, users, and content usage increases. Scaling capability is built into all portions of the system including user registration, user authentication, and usage logging.

##### 20 4.3.3 MAINTAINS DISTRIBUTED, LOCAL CONTROL OF THE USER BASE

Clickshare believes that centralization of the user base for the purpose of unified registration, profiling, or measurement is a potentially non-scalable, performance-limiting approach to user management. In the TVS system, the user base is managed at the "local" (publisher/service provider) level. This has technical, sociological and financial advantages.

25 One specific advantage is that the individual publisher/service provider is in control of the

customer billing relationship. The system thus presupposes multiple billing agents and requires no centralized database of user-specific demographic data.

#### 4.3.4. LOW IMPACT ON PERSONAL PRIVACY ISSUES

5 In the TVS model, independent content providers manage their own users, and cooperate to "pool" this user base. In addition to the technical reasons for this design decision, this allows the content providers to maintain the customer relationship, which in turn means that publishers are able to provide a richer (and even personalized) experience to their users. Further, locality of the user base offers a better chance that users' desire for privacy will be respected.

10 In providing the TVS service, Clickshare Corporation or its licensee maintains only transitory knowledge of any specific user, and even then, only by a user identification number (not by demographic or financial information).

#### 4.3.5. SUPPORTS A NUMBER OF USER BILLING MODELS

15 Clickshare believes that no single model of charging users will be completely effective in the diverse Internet environment. Further, it is not clear what all the options are, or how they might be administered in such a distributed environment. Consequently, TVS does not enforce a specific billing model, but rather enables a wide variety of models, including so-called "micro-transaction" or per-page billing.

#### 4.3.6. SUPPORT BOTH FIXED AND FLEXIBLE PROFILE INFORMATION 20 AND SERVICE CLASSES

25 Clickshare Corporation feels that as commercial use of the Internet increases, both users and providers will want different classes of service that can be offered to fit specific price requirements or usage demands. It is possible to develop very low-level class-of- and quality-of-service levels (and indeed this is being developed now by the Internet Engineering Task Force (IETF), and also possible to design solutions to fit each specific application or

each provider. TVS offers service and user class "infrastructure" as part of the TVS service, which itself is independent of application (though it is currently offered only through the World Wide Web "application") and provider. Because such classing is not yet well-defined in either engineering or marketing domains, Clickshare intends to work with its early partners on such development. The TVS service classing features are easily extended to provide a flexible growth path and accommodate change.

With this classing infrastructure in place, TVS also offers user profiling and preference specification capabilities, again with the same extension capabilities. These capabilities support many of the advantages that are to be gained from developing application-specific profiling, without the inherent incompatibilities of application specific design. application-specific profiling, without the inherent incompatibilities of application specific design.

#### 4.4. NO SPECIFIC PRIVACY MODEL ENFORCED

Clickshare realizes that being involved in the "authentication" universe means dealing with sensitive personal financial information. There is a lively debate on-going among privacy advocates and content providers (who use the sales of lists of such information to enhance their revenues).

The TVS model does not enforce a specific privacy model. The service itself operates by identifier numbers, not by names, and Clickshare Service Corp. -- on its own -- will not be able to correlate an ID with a person. However, nothing inherent in the TVS service specifically prevents a Publishing Member from making this correlation on his own through methods unrelated to the Clickshare service. It is possible within the design of TVS to offer a "Swiss-bank" type of "blind usage" for users that wish to pay for same. No such service is currently implemented.

As designed, TVS will be able to collect and aggregate content usage information and "localize" this information to a specific user-ID and provider-ID. This alone will go a long

way towards providing third party verification of use without direct reference to personal information.

#### 4.5. CHARGING PER-REQUEST A FEATURE NOT A REQUIREMENT

5 TVS is designed to handle charge-per-page billing in a cross-server settlement type model. However, the market may demand that such a service is not required (that is, not profitable). Therefore, TVS can operate as an authentication / verification engine without settlement. The presence of a 3rd party verifier has significant value even in this arena.

10 Further, in the charge-per-page model, TVS does not "force" publishers to bill the users themselves for each click. TVS merely enables each publisher to develop his own billing technique, with the service financed with any combination of ad subsidy, subscription and a la carte pricing.

#### 4.6. DESIGN FORESEES NEED FOR "TRUSTED THIRD PARTY"

15 Looking at the process diagrams shown in Figure 1 and 2, it is easy to imagine "cutting TVS out" and having HTTP servers talk among themselves. In a technical sense, with the protocol code in hand (and using almost the same API) this is possible. However, this then forces each individual HTTP server to both serve content requests, and serve all authentication requests whenever a user "roams" to any other service. This is distributed service, but possibly to the detriment of performance of every single server host. In the TVS model, as in so many other Internet client/server models, the individual server is relieved of the burden of performing the redundant function.

20 25 From a non-technical perspective, there is a benefit in allowing a trusted third-party to serve authentication and access logging functions. Having a trusted third party involved is a method of "neutral" verification of use, something already desired by advertisers (and already common on other communication/advertising media). In addition, all service providers know and trust the provider of the Clickshare/TVS authentication service, while not all service providers know or trust all other service providers from whom, in the absence of

the trusted third party, they would have to rely for user authentication and perhaps billing information as well.

### 5.1 THE MODEL

Clickshare Service Corp. feels that a distributed approach should be taken when developing a high volume service for the Internet. Indeed this is common practice and several existing scalable services, notably BIND-type domain name service, use this technique. The TVS model distributes a very large user base among the independent content providers, but offers a single service-point from which each provider can get information about any active user. The implementation of TVS parallels the BIND model wherein any individual host needs to contact only one "name server" for information, while the name servers themselves "cooperate" to manage user information (in a manner that is not of concern to the individual host). The TVS service is scalable from a single server host to a multi-host environment transparently to its clients.

Using the TVS model, individual publishers or service providers authenticate their own users, and then ask TVS to store the user's preference, pricing and service-class information in a "publicly accessible" place. In return, TVS provides an authentication token which is returned to the user (specifically, the user's browser). All subsequent access to any TVS-enabled service is governed by this token (non-TVS services are not affected). TVS validates the token on behalf of any individual service, and passes in return the user's profile and class information. When a server has provided service to a validated user, that server returns to TVS a record of the service provided. This record is used by TVS to generate a number of forms of usage information, particularly billing and settlement information. Periodically, this information is returned to all publishers.

### 5.2 TERMINOLOGY

25 A small number of terms are added to the lexicon by TVS.

### 5.2.1 Identifiers

To differentiate individual customers (called "User Members"), each is assigned an ID number, and similarly each content provider (called a "Publishing Member") is assigned a unique ID. Though an individual customer may have several accounts, each has a separate User Member ID (UID). Each of these IDs is assigned to a specific Publishing Member ID (PMID). This association is referred to as the user's "home". The TVS concept that each user has a "home" is important as it localizes the service to an individual Publishing Member and allows for graceful support of a number of error conditions that might occur with such a widely distributed user base.

### 5.2.2 Session

TVS introduces the notion of a "session" into the World Wide Web. Once a user is authenticated by his "home" Publishing Member, that Publishing Member provides user profile information to its TVS server, which returns an authentication token that is valid for a restricted period of time. Once given this token, the user can access any TVS-enabled HTTP server for the duration of validity without reauthentication. This time period is the "session". The user may directly end his session prior to the pre-determined time-out, but is not required to do so. Further, upon time-out-out, Clickshare can return the user to his "home" Publishing Member for re-authentication transparently. Thus, sessions can be concatenated as well.

### 5.2.3 User Profile

Publishing Members maintain a "user profile" of each User Member. This profile contains three types of information: "preference" information, "service class" information and, if desired, "pricing" information. Preference information is given by the user member, while service-class information and pricing information are provided by the Publishing Member. These types of information relate to the variety and quality of services offered by the Publishing Member, and each may affect the cost of that service. Some of the profile

information can be changed on a session basis, where other types can only be changed by the Publishing Member at fixed points.

5 At the start of each session, this profile information is passed to the TVS server when the HTTP server requests an authentication token for the user. The information is loaded by the TVS server into a Dynamic Session Database. When, during the session, any Publishing Member requests that TVS validate this authentication, TVS returns the profile information to that Publisher as part of the authentication. Thus, even though each user is "owned" by only one Publishing Member (the "home"), all Publishing Members have access to that user's profile information through TVS.

10 When a user's session expires, TVS drops that user's profile from the dynamic session database (though portions of it are logged in a service record). Thus, TVS has no permanent record of any individual user. The profile information passed to Clickshare by the Publishing Member's HTTP server does not include personal information that would allow a user to be identified by name or financial association. Clickshare indexes profile information by UID  
15 + PMID only.

#### 5.2.4 Authentication Token

20 When a user begins a session, his Publishing Member returns to him an authentication token (which he has received from TVS). The token contains information that identifies which TVS server originated the token, as well as information that identifies the associated user. The user (actually, his client browser) uses this token for all subsequent access to TVS-enabled publishers.

### 5.3. SERVICE COMPONENTS

25 The Clickshare Service has been designed as a distributed set of cooperating components that together provide an integrated user management environment. The initial focus of this environment is to provide micro-transaction settlement and audience measurement services to independent publishers and billing entities of all sizes and service

5 volumes. However, the environment can also be used to control access to services or intangible goods based upon attributes of the requesting user as revealed to the proposed service provider via the authentication token. These services need not have any monetary value. Some of these components run on computers operated by Clickshare Service Providers (at sites not affiliated with Clickshare Corporation). These are called the client side components. The rest, currently, run at Clickshare Corporation's own site. These are the server site components.

### 5.3.1. Server Side Components

#### Clickshare Authentication Service

10 This service authenticates users in real time allowing each user access to any Clickshare Service Provider without reauthentication for the duration of one session. This service is provided by a set of server machines distributed around the Internet for better fault tolerance and performance.

##### Components

- + Token Validation Service (TVS) server/daemon (tvsd)

#### Clickshare Logging Service

15 This service logs user transactions occurring at all Clickshare Service Providers sites, in real time. The major component of this service is the Logging Facility - a large database storing all transaction records for production billing. This facility can be operated behind a firewall, due to the design of the Facility interface server.

##### Components

- + Clickshare Logging Facility (SQL database) (mSQL)
- + facility server interface daemon (logd)

## Clickshare Settlement Service

The service "settles" accounts receivable / accounts payable activity among the Clickshare Service Providers on a periodic basis. It interfaces to the Logging Facility database environment in an "off-line" (non real-time) manner. Activity reports are generated for all parties. An interface to the Automated Clearinghouse (ACH) allows fully automated settlement.

### Components

- + settlement engine
- + interface to automated clearinghouse (ACH, Bank of Boston)

## Clickshare Billing Interface

This service provides periodic billing records and account summaries to each of the Clickshare Service Providers. It interfaces to the Logging Facility database environment through a set of billing procedures which themselves are tailored to interface with customer billing systems. Billing records are sent to the Service Providers via electronic mail. As an auxiliary capability, the Clickshare Billing Interface can generate user account update summaries upon request from the Service Providers.

### Components

- + billing record generator
- + billing report generator
- + session summary generator / remailer
- + interface to Visa/MC electronic merchant vendor service

## CLICKSHARE INTERCHANGE SERVICE

This service maintains a master database of Clickshare Service Provider and clickshare Publishing member IDs and periodically updates the slave databases on CSPs and CPMs' web

servers. Authentication and logging functions cannot be used by client servers unless the client server presents an ID which is contained in the master database.

### 5.3.2. Client Side Components

#### Clickshare Web Server

5 This service replaces the traditional HTTP (web) server on sites that participate in the Clickshare Service. It processes user requests for information/entertainment resources stored on the site. It interfaces to the Clickshare Authentication Service and the Clickshare Logging Service to manage, measure and account for access to these resources. Each Service Provider operates one of these servers.

#### 10 Components

- + Web Server deamon (httpd)
  - o NCSA HTTPD-based Web Server
  - o Apache-based Web Server
- + client interface to Clickshare Logging Facility (cs\_logd)

#### 15 Clickshare User Registration Environment

20 This service allows Service Providers to register users for the purposes of access control, service customization and billing. All user demographic and financial information (in addition to preference and service classing information) is stored in these databases at each Service Provider site. Users are authenticated locally from information stored in these databases, after which a subset of the stored information is provided to the Clickshare Authentication Service so that it can help all Service Providers recognize valid Clickshare users.

## Components

- + user registration facility (misc)
  - o interface modules for Clickshare Web Server
  - o stand-alone administration/maintenance programs
  - o HTML pages and forms for user interface

5

### 5.4. TVS "C" LANGUAGE APPLICATION PROGRAMMING INTERFACE

This section describes the TVS applications programming interface (API). This interface is written in the C programming language for wide applicability to most Web servers written in C, C++, or Objective C. Using this interface, programmers do not have to interact with the TVS wire protocol, or with the details of the internal data structures. Since Clickshare is still in its early development, this interface insulates programmers from the many changes that will occur in the low-level design. The TVS API was developed and written for the Unix(tm) operating system. The API uses widely-available and well-documented Unix(tm) features and functionality, such as Berkeley-type sockets. Clickshare intends that TVS be made available first in the Unix(tm) environment, but does not preclude porting the client API to other operating environments (notably, Microsoft Windows NT).

A better mode for implementation of the TVS model might be through the use of the Java programming language, or another language, to construct an applications mini-server which acts as a client to the TVS authentication and logging service. When an HTTP request is received by the Clickshare host seeking to vend a service or good, the miniserver would look for a TVS token and, if present, verify it and strip off the URL. Then the miniserver would connect to the waiting, unmodified, HTTP server deamon at the host site.. When the reply is served by the HTTP server, the miniserver would annotate the outgoing HTML code with the TVS token. The system administrator would configure the underlying HTTP server not to accept requests other than from the applications mini-server. The advantage to this approach would be its ability to work with any HTTP server without modification or significant reconfiguration.

10  
15  
20

25

## 5.5. SERVICE INTERFACE

### 5.5.1 Data Structures

5 The TVS Applications Program Interface (API) defines three data structures used as objects to the TVS library of function calls. Two of these can be accessed directly, but the third, the TVS\_TOKEN, is opaque. All three are treated as opaque since library function calls (described below) are provided to access the information inside where appropriate. The appellation "tv" in all the names comes from the pre-Marketing name for the Clickshare Service -- "Token Validation Service".

10 The following definitions are from the header file "tv\_client.h":

15 `typedef char *TVS_TOKEN;`  
`typedef char *TVS_PROFILE;`  
`typedef char *TVS_SERVER;`

### 5.5.2. Function Calls

15 Seven functions make up the interface to the TVS service.

15 These functions are also defined in the header file "tv\_client.h".

`/* tvs_initialize_service() is used to begin a session with TVS. This function returns a`  
`TVS_SERVER data structure that will be used in all subsequent service requests.`  
`*/`

20 `extern TVS_SERVER tvs_initialize_service();`

20 `/* These three requests are the bulk of the TVS service. They are called to obtain, validate,`  
`and invalidate user authentication tokens.`

`*/`

```
extern TVS_TOKEN  tvs_new_token(TVS_SERVER ts, TVS_PROFILE prof);

extern TVS_PROFILE tvs_validate_token(TVS_SERVER ts, TVS_TOKEN token, unsigned
long host_id);

extern int  tvs_invalidate_token(TVS_SERVER ts, TVS_TOKEN token, int reason);
```

5   /\* Occasionally, the TVS client may wish to get a human-readable version of the TVS server  
identifier for logging.  
\*/

```
extern char *  tvs_identify_tvs_server(TVS_SERVER ts);
```

10   When the TVS client wishes to gracefully end a session with its server, this function  
is called. Afterward, no requests will be served until another tvs\_initialize\_service() request  
is received.

```
extern void  tvs_drop_service(TVS_SERVER ts);
```

/\* if the TVS client exists badly from a session (machine crash or emergency signal), this  
function is called to assure that the client starts a clean session with a TVS server.

15   NOTE: the function tvs\_initialize\_service() calls this function prior to initializing any  
new session. It is "safe" to call this function even in the case that the client is certain there  
is no established session.  
\*/

```
extern void  tvs_invalidate_service(TVS_SERVER ts);
```

20   The following functions handle administrative and memory management aspects of  
using the TVS service.

/\* when Authentication Tokens are returned, they are allocated from dynamic memory. This function frees a dynamically-allocated token after it has been used.

\*/

extern void tvs\_destroy\_token(TVS\_TOKEN token);

5   /\* when TVS server errors occur, TVS tries to return  
   \* usable error information. Also, the API library manages local information used in error  
   \* reporting. These functions are used to access that error information.  
   \*/

10   extern void tvs\_perror(char \*string);  
      extern char \* tvs\_strerror();

/\* this definition and function allow the client to determine the status of a token.  
\*/

15   enum t\_error\_type {  
      TOKEN\_IS\_OK              = 0,  
      TOKEN\_IS\_GARBAGE        = 1,  
      TOKEN\_IS\_INVALID        = 2,  
      TOKEN\_TIMED\_OUT          = 3,  
      USER\_AT\_INVALID\_HOST   = 4,  
      NO\_TOKEN                 = 5  
20   };

extern int tvs\_token\_is\_error(int terr\_type);

/\*

\* if a token is not OK, it might be necessary to get the location of the user's home publisher.  
This is returned by TVS and available through this function.

\* /

```
extern char *    tvs_get_user_home();
```

## 5.6. PROFILE FUNCTIONS

When TVS receives a request to validate a valid token, it returns the profile of the user identified by that token. Though the `TVS_PROFILE` data structure is defined in the API, and data can be accessed directly, the following helper functions are provided. In general, this is the preferred access technique, because it is highly likely that the `TVS_PROFILE` structure will change.

```
/* these functions are for creating and "loading" a profile structure */
```

```
extern int    tvs_sizeof_profile();
extern TVS_PROFILE tvs_make_user_profile();

extern void  tvs_set_userid(TVS_PROFILE prof, unsigned long uid);
extern void  tvs_set_pmid(TVS_PROFILE prof, unsigned short pmid);
extern void  tvs_set_hostid(TVS_PROFILE prof, unsigned long hostid);
extern void  tvs_set_service_class(TVS_PROFILE prof, int class);
extern void  tvs_set_hostid(TVS_PROFILE prof, unsigned long hostid);
extern void  tvs_set_service_class(TVS_PROFILE prof, int class);
extern void  tvs_set_page_class_limit(TVS_PROFILE prof, int limit);
extern void  tvs_set_page_count_limit(TVS_PROFILE prof, int class);
extern void  tvs_set_service_priority(TVS_PROFILE prof, int priority);
extern void  tvs_set_pdac_flag(TVS_PROFILE prof, int flag);
extern void  tvs_set_customer_group(TVS_PROFILE prof, int group);
extern void  tvs_set_adv_context(TVS_PROFILE prof, int ac);
extern void  tvs_set_privacy1_flag(TVS_PROFILE prof, int flag);
```

/\* upon receipt of a TVS\_PROFILE - the acknowledgement of a token validation by TVS -  
information in the profile can retrieved using the following functions.

\*/

/\* this is a quick test that the data structure is not garbage \*/

5   extern int   tvs\_profile\_is\_valid(TVS\_PROFILE prof);

extern unsigned long   tvs\_get\_userid(TVS\_PROFILE prof);

extern unsigned short   tvs\_get\_pmid(TVS\_PROFILE prof);

extern unsigned long   tvs\_get\_hostid(TVS\_PROFILE prof);

extern int   tvs\_get\_service\_class(TVS\_PROFILE prof);

10

extern int   tvs\_get\_page\_class\_limit(TVS\_PROFILE prof);

extern int   tvs\_get\_page\_count\_limit(TVS\_PROFILE prof);

extern int   tvs\_get\_service\_priority(TVS\_PROFILE prof);

extern int   tvs\_get\_pdac\_flag(TVS\_PROFILE prof);

15

extern int   tvs\_get\_customer\_group(TVS\_PROFILE prof);

extern int   tvs\_get\_adv\_context(TVS\_PROFILE prof);

extern int   tvs\_get\_privacy1\_flag(TVS\_PROFILE prof);

/\* the contents of the profile can be displayed in human-readable form \*/

extern void   tvs\_show\_user\_profile(TVS\_PROFILE prof);

## 5.7. TVS PROTOCOL IMPLEMENTATION

20           The TVS wire protocol is simple, compact and invisible under the applications  
programming interface. Nonetheless, several important points should be discussed.

Please note that the API library interface allows Clickshare to perform a full frontal lobotomy  
on this protocol structure if required without changing the programming interface. Since the  
TVS service is new, Clickshare Service Corp. expects that the wire protocol (especially) will  
undergo significant change. Therefore, the wire protocol is not specified here.

## 5.8. RELIABLE DATAGRAMS FOR VERIFICATION

5 TVS is written using the User Datagram Protocol (UDP) on top of the Internet protocol (IP). The inherent "unreliability" of UDP is mitigated by the addition of a "reliability layer" which effects packet timeout and retransmission (from the client side). All protocol operations involve typically 2 UDP/IP datagrams (one in each direction, assuming no retransmissions), and a maximum of 4 datagrams under certain conditions of re-direction within the Clickshare service "back end" (invisible to the client).

10 The client side of TVS uses the BSD socket connect() system call to assure that only packets coming from its TVS server are accepted.

## 10 5.9. RELIABLE STREAM FOR USAGE LOGGING

15 When an HTTP server begins a service session with a TVS server, it opens and initializes a TCP/IP reliable data stream to a designated logging port on the TVS server. All service requests fulfilled by the HTTP server are logged back to TVS on this connection. The TVS HTTP reference server also performs the normal (local) logging.

20 15 The log record returned to TVS is, currently, compatible with the public domain server logs commonly used. Several pieces of additional information are added to each record, however, using an extensible key/value technique. This information assists TVS in the settlement process.

25 20 UDP has the advantage of being very lightweight. Since virtually all wire transactions contain under 1,000 bytes and are "atomic", the need for a "stateful" protocol like TCP/IP is greatly diminished. In the prototype, a reliability veneer is added over UDP to handle slow or unreliable networks. TCP/IP is used for logging the access records. This method was chosen because consideration is being given to "batching" log delivery (whereby the client saves all log records for "burst" delivery at slow server-utilization periods) and this will benefit from a stateful protocol. The current implementation of the logger is as a separate process (from the HTTP daemon) to facilitate this change if need be.

## 5.10. REQUEST AND TOKEN DELIVERY

5 The current interface to the TVS service is through the World Wide Web, specifically servers offering the HyperText Transfer Protocol. Using both Uniform Resource Locators (URLs) and HTTP Request/Response Headers, the HTTP server communicates with client programs ("web browsers"). Separately, the HTTP server communicates with the TVS service using the TVS protocol. Thus, TVS is not dependent upon HTTP for anything beyond delivery/return of the Authentication Token to/from the HTTP server. The TVS service itself (and the delivery interface in TVS "clients") will not be affected if new transfer protocols supersede HTTP.

10 The prototype implementation of TVS uses the Uniform Resource Locator (URL) to return the Authentication token (until a standard exists for transferring it in a Request/Response header). Therefore, TVS is dependent on the URL format, but not on the HyperText Markup Language (HTML), though HTML is the only widely-accepted text language that uses URLs. Should HTML be superseded by another markup language, TVS delivery is still possible. Experiments and early commercial work with Adobe Portable Document Format (PDF) and TeX's "Device Independent" format (DVI) have shown that these formats can accommodate URLs, and can be considered for hypertext applications.

15

## 5.11. TVS SERVER ARCHITECTURE

20 The TVS Service is actually a set of services available to clients through an object-oriented modular architecture. The details of how these services are provided are "hidden" within this object structure and applications programming interface.

25 Currently, the Clickshare "session management" (user authentication and validation) services comprise one service, the access logging another, and the settlement service yet another. Session management and access logging services are "real-time", designed to handle and scale to a very large load with minimum impact on service latency. Settlement is an off-line non-real time process which does not effect the provision of the real-time services.

## 5.12. THE TOKEN VALIDATION SERVICE (TVS)

5 The Token Validation Service (TVS) handles user session management. After a TVS-enabled HTTP server authenticates a new user, it passes that user's profile information to Clickshare along with a request for a new authentication token. Thereafter, for the duration of that user's session, the Clickshare TVS server "validates" the user whenever the user presents a URL request to any TVS-enabled HTTP server.

10 The TVS server maintains a Dynamic Session Database (short-lived) of active sessions, indexed by user identification number, "home" publisher affiliation, and the user's host IP address. Among the data contained in the Dynamic Session Database are:

15

- Alpha-numeric identifying number of the user
- User-owning publishing-member number (Clickshare Service Provider)
- Session number
- Current number of authentications (cumulative)
  
- User service parameters including:
  - Parental control flag (ON/OFF)
  - Full ads / links only / no ads
  - Pricing query threshold
  - Service-class designator (price markup value)
  - Session start time
  - Topical information preferences (if "open")
  - Age, sex, income, demographic profile (if "open")

20

Upon completion of a session, this information is logged to provide a record of the session which will be used to accompany billing and settlement information. No permanent knowledge of a user, or a specific session, is retained by this server.

25 The TVS server databases are very straight-forward, but very fast, hash/value tables which use GDBM from the Free Software Foundation. This technology scales to more than

one million entries (one million active sessions), and offers features for restart and fast-startup. Virtually no "database management" functionality is required in this application since no permanent knowledge is stored here.

5 Since the TVS service is provided on machines separate from the HTTP servers, there is a possibility that either machine failure or network outage may make the service unavailable temporarily. In such cases, the HTTP server will issue itself a "restart" which will attempt to reconnect the server to another TVS server on another part of the network. Users with active sessions will have to re-authenticate with their home publisher, but this is transparent given graceful handling by the TVS client web server and caching of 10 username/password in most browsers.

#### 5.12.1. TVS Process Model

15 The use of UDP/IP as the underlying wire protocol and fast hash/value tables for the authentication/validation portion of the TVS service allows the use of a single process for the TVS server, except under specific circumstances (see below). However, the TVS server software design allows for future migration to a "master and pre-allocated slave" model if the load outstrips the single process model.

#### 5.12.2. TVS Scalable Multi-Server Environment

20 The HTTP server's actions of initiating a session with TVS and asking TVS to generate a new Authentication Token require communication with only one TVS server. However, a single TVS server cannot permanently serve more than a small number of clients. Therefore, the service needs to scale (in terms of addition of servers) to provide service "bandwidth". Once such a multi-server environment is introduced, there will be 25 inter-server communication -- in addition to client/server communication -- for the actions of validating and invalidating authentication tokens. The TVS wire protocol and server software are designed for this multi-server environment.

It is anticipated that the Clickshare Service will include a Clickshare Interchange Service [See Figure 2, Service Model for Multiple Authentication/Logging Licensees], which will maintain a master database of IDs for Clickshare Publishing Members and Clickshare Service Providers and this master database will be periodically distributed in a secure manner to slave databases maintained by multiple Clickshare Authentication and Logging Servers (CALS). These CALS may be independently licensed and owned but will, using the TVS protocol, have the technical capability to authenticate each others end users and relay real-time log reports for settlement. In effect, each CALS functions not only as a server to CSP and CPM clients, but also as a server to its peer CALS. Thus it is relatively transparent to each CALS whether it receive a token validation request or a log report from a CSP, a CPM or another CALS.

The TVS client-side (HTTP server) architecture is designed so that on start-up (or soft restart) the HTTP server searches for it's service-provider from a local list of options. If, due to network or service host failure, the Clickshare authentication or logging facilities become unavailable, the HTTP server will restart and search for a new (available) service host. Since authentication tokens handed out by the now-dead TVS service are invalid, users will be returned to their home publishers to re-authenticate. This condition is handled gracefully through HTTP redirects, such that the user may not even see this condition (depends on the user's browser type). Independent of the number of TVS servers, and independent of the number of HTTP servers being served, at most two TVS servers will be involved in validating or invalidating a specific authentication token. Thus, there will be at most four (4) packet transmissions required for the most-commonly occurring activity (validation), assuming no retransmissions.

### 5.13. THE TVS LOGGING FACILITY

All information transactions that occur on all TVS-enabled HTTP servers are logged to the Clickshare Logging Facility. This service resides on the same computers as the authentication service such that each HTTP server selects one computer to provide both services upon startup.

5 The logging service is provided on the client (HTTP) side by a separate process accessible through a Unix-domain UDP socket. This process maintains a reliable TCP/IP connection to the TVS Logging Facility. The HTTP server accesses the logging process as part of the normal (local) logging activity done by the server. Clickshare-specific information is added only to the log information intended for the Clickshare logging facility (not the local log).

10 On the server (logging facility) side, a master logging process spawns a slave process for each connected client. Each of these slave processes opens a connection to the Facility's log database manager. This database manager can reside on the log server machine or on another machine (accessed through a network socket). The Facility's databases are organized such that as transaction records are returned they are "filtered" by "owning publisher". Thus all records for one user reside in one database - that of his "home" publisher. These log databases are updated in real-time. The Facility's database manager can "dump" these databases into the offline settlement service at any appropriate frequency (daily, weekly, monthly) depending on load.

15 The logging processes use SQL to "talk to" the Facility's database manager. Currently, the logging processes are C language programs which use our own API to insulate the programs from the specifics of our vendor's interface to its SQL database manager.

20 The service envisions that some end users will wish to query, through an HTML-forms interface or otherwise, the filtered server log database, to determine charges which have been applied to the user's Clickshare ID number since the last settlement to a billing/credit facility. This application will require a separate real-time application which is aware of the pricing rules being applied by the end user's Clickshare Service Provider -- and to which the end-user has "subscribed" -- providing the application with the end user's 25 globally unique Clickshare ID number. When queried, the application can parse from the owning service provider's server active log database all records associated with access by the querying user's Clickshare ID number and present them in an HTML form for review. The purpose of this application is to permit users to gauge their rate or resource usage, or permit

Clickshare Service Providers to apply session-based charges against billing or credit facilities  
-- including smart cards -- during periods between system settlement.

#### 5.13.1. CLICKSHARE-ENHANCED LOG FORMAT

5 The TVS client transmits to the server (logging facility) side records of each access  
in an enhanced Common Log Format. Seven pieces of information are provided in the  
Common Log Format:

- fully qualified domain name (or dotted decimal IP address) of the client
- rfc931 user
- auth user
- date in dd/mm/yyyy:hr:mn:se -0XXX format (where 0XXX" is hours  
from GMT)
- the request (a quoted string featuring method + URL filepart)
- the HTTPD status code
- the number of bytes transmitted to the client

10 In addition, the TVS client transmits the following Clickshare-specific information:

15 -- content server ID (cs\_contentpmid) -- A globally unique ID number identifying the  
company which served the content to the user. Clickshare Service Corp. maintains a map of  
ID numbers to company names and contact addresses.

20 -- page class (cs\_pageclass) -- A numeric identifier for the value of the page served.  
The value is used as a lookup into a table of currency-denominated values which are used to  
price the page.

-- user ID (cs\_userid) -- A user identifier, unique to each Clickshare service or content  
provider, that identifies the user within that provider's site.

-- home publisher ID (cs\_homepmid) -- A globally unique ID number identifying the company which maintains the financial relationship with the user (user ID) for billing purposes.

5 -- session ID (cs\_sessionID) -- An identifier for an activity session by a user. A *session* is a defined period of time during which an authentication token is valid. The length of a session can be requested by the user, or set by the home provider, upon startup). Sessions may be concatenated in time, but sessions cannot overlap. Session IDs are unique to each publisher for a period of about eight months.

10 -- customer group (cs\_custgroup) -- A numeric identifier for the customer's local group. Two groups are global within Clickshare: Group 1, the default standard group and Group 15, the "testdrive" group. All other values are set locally by the home publisher for his own reference.

15 -- service class (cs\_serviceclass) -- A coded numeric identified for special service classing. Service classes may be related to markup ratios for retail pricing or may specific the types of services or goods which the user is authorized to acquire or receive.

-- flags (cs\_flags) -- A coded numeric identifier which concatenates all the user-preference flag information (on/off flags) for this session. These preference flags relate to user privacy, parental-control (content selection) and other features and part of the "contract" between the user and the user's Clickshare Service Provider.

20 Other open data blocks are designed to carry releasable demographics and topical preferences, or other metrics, depending upon the requirements of Clickshare service members.

#### 5.14. USER REGISTRATION ENVIRONMENT

25 The prototype implementation of the Clickshare/TVS service provides a User Registration Environment as a distinct application which is compatible with the TVS client web server. Upon a user's first attempt to access resources within the TVS environment the

user is redirected to the User Registration Service. The user is requested to enter their name, choose a password, and provide certain demographic and account information, including credit-card data. In some versions of the User Registration Service, the user is also requested to enter preference information about Service Class, Query Threshold, Privacy and Parental Control and Content Preferences. The prototype implementation does not provide for real-time verification of credit card information, but this should be provided in commercial implementations.

#### 5.15. SETTLEMENT OF USAGE DATA

10 The prototype implementation of the Clickshare/TVS service provides a Settlement Service as a distinct database-management application which operates in conjunction with the TVS logging engine. This Settlement Service shorts records of user access to resources by Service Provider and by user within Service Provider and prepares the records for batch deliveries to the individual user's Service Provider. The Settlement Service also outputs charge records aggregated by Service Provider in a format which can be accepted by gateways to the U.S. banking industry's Automated Clearing House (ACH) service for electronic debiting and crediting of Service Provider and Publishing Member banking accounts. Finally, the Settlement Service outputs charge records aggregated by end user within Service Provider to a Billing Service in a format specific to the most common PC-based program for application of charges to credit-card gateways. Finally, Clickshare 15 contemplates an interface for outputting individual, aggregated user charges in a format which can be employed by resellers of telephone company billing services to apply transactional charges to telephone bills.

#### 20 5.16. DIFFERENTIAL CHARGING PER RECORD ("MARKUP RATIO")

25 A unique feature of the TVS protocol is the provision for a field which permits differential charging for information sale between an information owner/publisher and various Service Providers ("user owners"). Each logged charge record includes a data field carrying the "[Page] Value Class" of the content or service vended by the vending provider site (publisher). The Value Class is a numeric value assigned by the vending provider from a

common table of values known and agreed on a system-wide basis. This Value represents the "wholesale royalty" which the vending site will receive at settlement from Clickshare/TVS, less a transaction fee withheld and charged by Clickshare/TVS.

5        However, the Service Provider who maintains account records of the end user and is responsible for billing the end user, may wish to apply a retail markup to the Value Class expected by the vending publisher. In the TVS Service, this retail markup is known as the "markup ratio." The "markup ratio" is derived from the Service Class identifier, and is among the user preference attributes contained in the token which moves about the service following initial authentication of the user for a TVS session. The operator of the TVS authentication and logging service does not use this field in any way. However, the vending publisher may need to know the Markup Ratio being applied by the Service Provider in order to "show" the price of a service or resource to the end-user prior to or during the sale process.

10       The markup ratio allows the vending publisher to provide an instantaneous "price tag", including any retail markup, to the requesting user, prior to sale. Because the markup ratio is set by the Service Provider, two users from two different Service Providers may pay a different retail price for the same resource provided at the same time by a vending publisher. In effect, the same good may show different "price tags" simultaneously to different users. This feature is analogous to the operation of the wholesale-retail marketplace in the physical world, where consumers may find widely varying retail prices for the same physical product at different stores as a result of different marketing and cost structures of each retailer. For example, one appliance store might sell the Brand A toaster for \$29.95. A competing appliance store might sell the identical Brand A toaster for \$25.95. Both purchased the toaster from a wholesaler for \$14.95. Under the Clickshare Service, the wholesaler must be able to "show" both prices depending which "store" the consumer is inquiring from.

#### 5.17. PRIVATE-KEY ENCRYPTION USED

25       In the current implementation of the TVS service, almost no encryption is used. The only transacted item that is encrypted is the authentication token which travels along with

each user request. This token is issued by the Clickshare/TVS authentication server, and only that server needs to "decrypt" it. All other parties (HTTP servers, and other TVS servers) treat the token as "opaque". Since only the originating TVS server will view the contents of the authentication token, a "private key" encryption algorithm can be used. That private key is stored on the authentication server which originates the token, and remains valid only for the duration of that user's session.

Currently, TVS uses the IDEA encryption algorithm with a 128-bit key. Since the encrypted data is only valid for the length of one session, and further since the encrypted data has no "enduring value" that warrants use after the session (e.g., credit card number or index), a 128-bit key is sufficient to protect this data for the duration of the session.

In the future, experience may require that the communication channel between the HTTP server and the Clickshare authentication and logging facilities be encrypted. Since the nature and format of these communication channels is "hidden" in the C programming language API, addition of encryption is totally transparent to both the authentication and logging processes (there are likely to be externally visible additions to configuration files, however).

#### 5.18. OTHER MODES FOR TRANSFERRING TOKEN

To transfer tokens, we selected the query-string presentation technique because it is completely browser-neutral (a trait which appears necessary to get TVS widely used initially). However, this technique makes it difficult for the HTTP server to calculate the Content-Length in advance, and makes it easy for any user to cut-and-paste a valid token to an invalid request. Further, the "one Token per session" notion is subject to wire snooping, where a snooper can grab the Token, and use it in combination with a falsified IP address to gain unauthorized access.

A better technique is to put the token in an Authentication request/response header. This technique is suggested in the recent IETF "Digest Authentication" draft, and a demonstration of this will be present in the early TVS service. If web browsers and other

web clients widely adopt this technique (we think this is reasonably likely), this will limit certain forms of token theft.

5 An even better technique, which would require special additional software on all web clients, is the combination of the Digest Authentication technique with a one-time password technique (as described in another IETF proposal from Bellcore). Using this technique, the TVS service could authorize a limited number of access attempts (say, 150) after which time the session is invalid. At each use, the client sends a one-time password, the "base" for which is pre-agreed (such that the server is able to correctly identify the "next" in a series).  
10 This has the additional advantage that authentication can be somewhat de-coupled from IP address (as is the case in the current TVS model).

15 Recently, the IETF's HTTP Working Group offered an Internet Draft entitled "A Proposed Extension to HTTP: Digest Access Authentication" for discussion by the standards community. Though intended for a single-publisher relationship, this authentication method can be used by TVS-enabled HTTP servers as well, without modification (that is, "as proposed"). A prototype implementation of this technique will be in the reference implementation, though the proposal is likely to undergo a number of modifications before its acceptance.

#### 5.19. STATUS OF PUBLIC-KEY ENCRYPTION

20 Both of the above alternatives can make use of private key encryption -- which is (largely) unencumbered by legal concerns.

25 There is tremendous turmoil surrounding security these days, particularly as relates to passing credit information over the wire. Further, there is much debate over the legality and exportability of virtually all "public key" encryption techniques. Until there is clarity on these issues - clarity that will allow developers to use security techniques without fear of arrest or hefty legal defense expenses - Clickshare Service Corp. feels it is not beneficial to become involved in this arena for the area of commerce TVS intends to pursue. In the TVS model, no credit card information passes over the wire. Thus, the only gain in obtaining a

token is access to low-value content for a limited period of time. This is quite different from theft of credit card information which can result in thefts of very significant dollar value over an unlimited period of time. Therefore, a number of current, unencumbered, techniques can be employed.

Clickshare Service Corp. does not discount the value of security or any of the high-security protocols extant or imagined. However, there needs to be match of security to exposure. This match is largely missing from the current debate. We believe that the TVS model can and will benefit from new work, and will be able to adapt missing from the current debate. We believe that the TVS model can and will benefit from new work, and will be able to adapt to these new tools.

#### 5.20. USE OF EXTENSIBLE MARKUP LANGUAGE TO ADDRESS NETWORK CACHING OF CONTENT

(Please refer to Figure 2 for terminology lexicon)

The core architecture of TVS assumes that the "owner" of resources served to end-users corresponds to the Publishing Member or Service Provider on whose server the resource resides. By making this assumption, it is possible to automatically allocate royalty payments for chargeable resources according to the PmbMbr/SrvcProviderID of the vending content server with reference to the contents of the resource itself.

However, a variety of schemes are being proposed for replicating or caching original content on multiple physical servers. These proposals are intended to reduce bandwidth bottlenecks on the Internet by dispersing popular content to many locations. This raises the possibility that a resource "owned" by CPM<sub>a</sub>, for example, might come to reside on the physical network server of CPM<sub>b</sub>. In such a case, it is CPM<sub>a</sub> who is entitled to a royalty payment when the resource is physically served by CPM<sub>b</sub>.

At the same time, the near-standardization of Extensible Markup Language (XML) presents an opportunity to create the notion of a Universal Document Identifier (UDI) which

would be embedded in XML-marked-up documents or resources. This UDI could include six numeric components which, taken together, constitute a globally unique resource identifier.

5.20.1. The creation date/time in YYYYMMDDHHMMSS format.

5 5.20.2. The PubMbrID of the resource owner entitled to receive royalty payments, until the expiration date (see 5.20.4 below).

5.20.3. A Digital Object Identifier (DOI) in the syntax and form as proposed by the International DOI Foundation [see <http://www.doi.org/>] or, if DOI fails to become a publisher standard, in the format of a Uniform Resource Name(URN) as proposed by the Internet Engineering Task Force.

10 5.20.4. A expiration-date in YYYYMMDD format after which the document is prohibited from being served directly. Instead, the CPM server must redirect the requesting CM user to the address derived from the Digital Object Identifier.

15 5.20.5. A royalty payment which becomes due to the resource/document originator whenever a copy is served. This field would include both a designation of the currency and the payment value.

20 5.21. The initial version of TVS contains no facility for examining the contents of documents for data fields of interest. Subsequent versions should include the ability for the vending CPMx's server to review a resource prior to serving to determine if an XML-coded Universal Document Identifier is present. If present, the vending CPM should be required to use the PubMbrID and price information from the XML markup in place of default values when submitted a log report of the document purchase to its CSPx.

5.22. PROPOSED IMPLEMENTATION OF "DEAD MAN'S SWITCH" IN TVS  
PROTOCOL

The prototype implementation of TVS is designed to accommodate the failure of a CALS to respond to an authentication request from a CPM, either because of a network interruption or because the CALS itself is "down" (not operating). In such a circumstance, it is desirable for the CPM to be able to continue to reply affirmatively to HTTP requests for resources from end-user CMs. Later versions of TVS will permit the CPM to go ahead and serve resources, storing the log reports in a temporary, local database. On a periodic basis, the CALS will poll all affiliated CPMs and "pull" these log reports for subsequent aggregation and settlement. The business rules of the TVS operating entities will determine who bears financial responsibility for information resources served during the period when real-time authentication of user tokens is unavailable.

5.23. CREDIT MANAGEMENT SERVICE OPTION

Versions of TVS will include a Credit Management Service (CMS) feature which accommodates the need for the establishment of a financial "credit limit" set by the CSP of the CM end user. This will be accomplished through the addition of a field to the user-profile data (or the extended use of pricing query threshold field) which is stored in the CSP's dynamic session database when the CM user initiates a session [See 5.13.1, above]. The operation of the CMS is as follows:

5.23.1. CM user initiates a session request and user's CSP provide CALS with data for dynamic session database. Included in data is a value, Credit(x), established by the CSP in a reference currency supported by the relevant CALS. This figure might be (a) a percentage of the stored value in the CM user's "smart card", or (b) in an account linked to the user's "debit card" or (c) an amount established by whatever entity is working with the CSP to extend financial credit to the CM user.

5.23.2. Each time the CALS receives an authentication request from a CPM who wishes to vend information to the CM user, CALS responds with the user profile data from its dynamic session database, including the Credit(x) value.

5 5.23.3. The vending CPM's server views the Credit(x) value and computationally compares it against the value of the resource to be vended, incorporating any markup ratio inherent in the cs\_serviceclass field value also received from the CAPS dynamic session database. The CPM server knows the cs\_pageclass of the resource to be served either by reference to its physical directory location or by reference to a Universal Document Identifier field embedded within the resource. If Credit(x) is less than the monetary value of cs\_pageclass times the 10 markup ratio, CPM's server refuses to serve the resource. If Credit(x) is greater, and other service rules are met, the CPM server returns the resource to CM user.

10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 105 110 115 120 125 130 135 140 145 150 155 160 165 170 175 180 185 190 195 200 205 210 215 220 225 230 235 240 245 250 255 260 265 270 275 280 285 290 295 300 305 310 315 320 325 330 335 340 345 350 355 360 365 370 375 380 385 390 395 400 405 410 415 420 425 430 435 440 445 450 455 460 465 470 475 480 485 490 495 500 505 510 515 520 525 530 535 540 545 550 555 560 565 570 575 580 585 590 595 600 605 610 615 620 625 630 635 640 645 650 655 660 665 670 675 680 685 690 695 700 705 710 715 720 725 730 735 740 745 750 755 760 765 770 775 780 785 790 795 800 805 810 815 820 825 830 835 840 845 850 855 860 865 870 875 880 885 890 895 900 905 910 915 920 925 930 935 940 945 950 955 960 965 970 975 980 985 990 995 1000 1005 1010 1015 1020 1025 1030 1035 1040 1045 1050 1055 1060 1065 1070 1075 1080 1085 1090 1095 1100 1105 1110 1115 1120 1125 1130 1135 1140 1145 1150 1155 1160 1165 1170 1175 1180 1185 1190 1195 1200 1205 1210 1215 1220 1225 1230 1235 1240 1245 1250 1255 1260 1265 1270 1275 1280 1285 1290 1295 1300 1305 1310 1315 1320 1325 1330 1335 1340 1345 1350 1355 1360 1365 1370 1375 1380 1385 1390 1395 1400 1405 1410 1415 1420 1425 1430 1435 1440 1445 1450 1455 1460 1465 1470 1475 1480 1485 1490 1495 1500 1505 1510 1515 1520 1525 1530 1535 1540 1545 1550 1555 1560 1565 1570 1575 1580 1585 1590 1595 1600 1605 1610 1615 1620 1625 1630 1635 1640 1645 1650 1655 1660 1665 1670 1675 1680 1685 1690 1695 1700 1705 1710 1715 1720 1725 1730 1735 1740 1745 1750 1755 1760 1765 1770 1775 1780 1785 1790 1795 1800 1805 1810 1815 1820 1825 1830 1835 1840 1845 1850 1855 1860 1865 1870 1875 1880 1885 1890 1895 1900 1905 1910 1915 1920 1925 1930 1935 1940 1945 1950 1955 1960 1965 1970 1975 1980 1985 1990 1995 2000 2005 2010 2015 2020 2025 2030 2035 2040 2045 2050 2055 2060 2065 2070 2075 2080 2085 2090 2095 2100 2105 2110 2115 2120 2125 2130 2135 2140 2145 2150 2155 2160 2165 2170 2175 2180 2185 2190 2195 2200 2205 2210 2215 2220 2225 2230 2235 2240 2245 2250 2255 2260 2265 2270 2275 2280 2285 2290 2295 2300 2305 2310 2315 2320 2325 2330 2335 2340 2345 2350 2355 2360 2365 2370 2375 2380 2385 2390 2395 2400 2405 2410 2415 2420 2425 2430 2435 2440 2445 2450 2455 2460 2465 2470 2475 2480 2485 2490 2495 2500 2505 2510 2515 2520 2525 2530 2535 2540 2545 2550 2555 2560 2565 2570 2575 2580 2585 2590 2595 2600 2605 2610 2615 2620 2625 2630 2635 2640 2645 2650 2655 2660 2665 2670 2675 2680 2685 2690 2695 2700 2705 2710 2715 2720 2725 2730 2735 2740 2745 2750 2755 2760 2765 2770 2775 2780 2785 2790 2795 2800 2805 2810 2815 2820 2825 2830 2835 2840 2845 2850 2855 2860 2865 2870 2875 2880 2885 2890 2895 2900 2905 2910 2915 2920 2925 2930 2935 2940 2945 2950 2955 2960 2965 2970 2975 2980 2985 2990 2995 3000 3005 3010 3015 3020 3025 3030 3035 3040 3045 3050 3055 3060 3065 3070 3075 3080 3085 3090 3095 3100 3105 3110 3115 3120 3125 3130 3135 3140 3145 3150 3155 3160 3165 3170 3175 3180 3185 3190 3195 3200 3205 3210 3215 3220 3225 3230 3235 3240 3245 3250 3255 3260 3265 3270 3275 3280 3285 3290 3295 3300 3305 3310 3315 3320 3325 3330 3335 3340 3345 3350 3355 3360 3365 3370 3375 3380 3385 3390 3395 3400 3405 3410 3415 3420 3425 3430 3435 3440 3445 3450 3455 3460 3465 3470 3475 3480 3485 3490 3495 3500 3505 3510 3515 3520 3525 3530 3535 3540 3545 3550 3555 3560 3565 3570 3575 3580 3585 3590 3595 3600 3605 3610 3615 3620 3625 3630 3635 3640 3645 3650 3655 3660 3665 3670 3675 3680 3685 3690 3695 3700 3705 3710 3715 3720 3725 3730 3735 3740 3745 3750 3755 3760 3765 3770 3775 3780 3785 3790 3795 3800 3805 3810 3815 3820 3825 3830 3835 3840 3845 3850 3855 3860 3865 3870 3875 3880 3885 3890 3895 3900 3905 3910 3915 3920 3925 3930 3935 3940 3945 3950 3955 3960 3965 3970 3975 3980 3985 3990 3995 4000 4005 4010 4015 4020 4025 4030 4035 4040 4045 4050 4055 4060 4065 4070 4075 4080 4085 4090 4095 4100 4105 4110 4115 4120 4125 4130 4135 4140 4145 4150 4155 4160 4165 4170 4175 4180 4185 4190 4195 4200 4205 4210 4215 4220 4225 4230 4235 4240 4245 4250 4255 4260 4265 4270 4275 4280 4285 4290 4295 4300 4305 4310 4315 4320 4325 4330 4335 4340 4345 4350 4355 4360 4365 4370 4375 4380 4385 4390 4395 4400 4405 4410 4415 4420 4425 4430 4435 4440 4445 4450 4455 4460 4465 4470 4475 4480 4485 4490 4495 4500 4505 4510 4515 4520 4525 4530 4535 4540 4545 4550 4555 4560 4565 4570 4575 4580 4585 4590 4595 4600 4605 4610 4615 4620 4625 4630 4635 4640 4645 4650 4655 4660 4665 4670 4675 4680 4685 4690 4695 4700 4705 4710 4715 4720 4725 4730 4735 4740 4745 4750 4755 4760 4765 4770 4775 4780 4785 4790 4795 4800 4805 4810 4815 4820 4825 4830 4835 4840 4845 4850 4855 4860 4865 4870 4875 4880 4885 4890 4895 4900 4905 4910 4915 4920 4925 4930 4935 4940 4945 4950 4955 4960 4965 4970 4975 4980 4985 4990 4995 5000 5005 5010 5015 5020 5025 5030 5035 5040 5045 5050 5055 5060 5065 5070 5075 5080 5085 5090 5095 5100 5105 5110 5115 5120 5125 5130 5135 5140 5145 5150 5155 5160 5165 5170 5175 5180 5185 5190 5195 5200 5205 5210 5215 5220 5225 5230 5235 5240 5245 5250 5255 5260 5265 5270 5275 5280 5285 5290 5295 5300 5305 5310 5315 5320 5325 5330 5335 5340 5345 5350 5355 5360 5365 5370 5375 5380 5385 5390 5395 5400 5405 5410 5415 5420 5425 5430 5435 5440 5445 5450 5455 5460 5465 5470 5475 5480 5485 5490 5495 5500 5505 5510 5515 5520 5525 5530 5535 5540 5545 5550 5555 5560 5565 5570 5575 5580 5585 5590 5595 5600 5605 5610 5615 5620 5625 5630 5635 5640 5645 5650 5655 5660 5665 5670 5675 5680 5685 5690 5695 5700 5705 5710 5715 5720 5725 5730 5735 5740 5745 5750 5755 5760 5765 5770 5775 5780 5785 5790 5795 5800 5805 5810 5815 5820 5825 5830 5835 5840 5845 5850 5855 5860 5865 5870 5875 5880 5885 5890 5895 5900 5905 5910 5915 5920 5925 5930 5935 5940 5945 5950 5955 5960 5965 5970 5975 5980 5985 5990 5995 6000 6005 6010 6015 6020 6025 6030 6035 6040 6045 6050 6055 6060 6065 6070 6075 6080 6085 6090 6095 6100 6105 6110 6115 6120 6125 6130 6135 6140 6145 6150 6155 6160 6165 6170 6175 6180 6185 6190 6195 6200 6205 6210 6215 6220 6225 6230 6235 6240 6245 6250 6255 6260 6265 6270 6275 6280 6285 6290 6295 6300 6305 6310 6315 6320 6325 6330 6335 6340 6345 6350 6355 6360 6365 6370 6375 6380 6385 6390 6395 6400 6405 6410 6415 6420 6425 6430 6435 6440 6445 6450 6455 6460 6465 6470 6475 6480 6485 6490 6495 6500 6505 6510 6515 6520 6525 6530 6535 6540 6545 6550 6555 6560 6565 6570 6575 6580 6585 6590 6595 6600 6605 6610 6615 6620 6625 6630 6635 6640 6645 6650 6655 6660 6665 6670 6675 6680 6685 6690 6695 6700 6705 6710 6715 6720 6725 6730 6735 6740 6745 6750 6755 6760 6765 6770 6775 6780 6785 6790 6795 6800 6805 6810 6815 6820 6825 6830 6835 6840 6845 6850 6855 6860 6865 6870 6875 6880 6885 6890 6895 6900 6905 6910 6915 6920 6925 6930 6935 6940 6945 6950 6955 6960 6965 6970 6975 6980 6985 6990 6995 7000 7005 7010 7015 7020 7025 7030 7035 7040 7045 7050 7055 7060 7065 7070 7075 7080 7085 7090 7095 7100 7105 7110 7115 7120 7125 7130 7135 7140 7145 7150 7155 7160 7165 7170 7175 7180 7185 7190 7195 7200 7205 7210 7215 7220 7225 7230 7235 7240 7245 7250 7255 7260 7265 7270 7275 7280 7285 7290 7295 7300 7305 7310 7315 7320 7325 7330 7335 7340 7345 7350 7355 7360 7365 7370 7375 7380 7385 7390 7395 7400 7405 7410 7415 7420 7425 7430 7435 7440 7445 7450 7455 7460 7465 7470 7475 7480 7485 7490 7495 7500 7505 7510 7515 7520 7525 7530 7535 7540 7545 7550 7555 7560 7565 7570 7575 7580 7585 7590 7595 7600 7605 7610 7615 7620 7625 7630 7635 7640 7645 7650 7655 7660 7665 7670 7675 7680 7685 7690 7695 7700 7705 7710 7715 7720 7725 7730 7735 7740 7745 7750 7755 7760 7765 7770 7775 7780 7785 7790 7795 7800 7805 7810 7815 7820 7825 7830 7835 7840 7845 7850 7855 7860 7865 7870 7875 7880 7885 7890 7895 7900 7905 7910 7915 7920 7925 7930 7935 7940 7945 7950 7955 7960 7965 7970 7975 7980 7985 7990 7995 8000 8005 8010 8015 8020 8025 8030 8035 8040 8045 8050 8055 8060 8065 8070 8075 8080 8085 8090 8095 8100 8105 8110 8115 8120 8125 8130 8135 8140 8145 8150 8155 8160 8165 8170 8175 8180 8185 8190 8195 8200 8205 8210 8215 8220 8225 8230 8235 8240 8245 8250 8255 8260 8265 8270 8275 8280 8285 8290 8295 8300 8305 8310 8315 8320 8325 8330 8335 8340 8345 8350 8355 8360 8365 8370 8375 8380 8385 8390 8395 8400 8405 8410 8415 8420 8425 8430 8435 8440 8445 8450 8455 8460 8465 8470 8475 8480 8485 8490 8495 8500 8505 8510 8515 8520 8525 8530 8535 8540 8545 8550 8555 8560 8565 8570 8575 8580 8585 8590 8595 8600 8605 8610 8615 8620 8625 8630 8635 8640 8645 8650 8655 8660 8665 8670 8675 8680 8685 8690 8695 8700 8705 8710 8715 8720 8725 8730 8735 8740 8745 8750 8755 8760 8765 8770 8775 8780 8785 8790 8795 8800 8805 8810 8815 8820 8825 8830 8835 8840 8845 8850 8855 8860 8865 8870 8875 8880 8885 8890 8895 8900 8905 8910 8915 8920 8925 8930 8935 8940 8945 8950 8955 8960 8965 8970 8975 8980 8985 8990 8995 9000 9005 9010 9015 9020 9025 9030 9035 9040 9045 9050 9055 9060 9065 9070 9075 9080 9085 9090 9095 9100 9105 9110 9115 9120 9125 9130 9135 9140 9145 9150 9155 9160 9165 9170 9175 9180 9185 9190 9195 9200 9205 9210 9215 9220 9225 9230 9235 9240 9245 9250 9255 9260 9265 9270 9275 9280 9285 9290 9295 9300 9305 9310 9315 9320 9325 9330 9335 9340 9345 9350 9355 9360 9365 9370 9375 9380 9385 9390 9395 9400 9405 9410 9415 9420 9425 9430 9435 9440 9445 9450 9455 9460 9465 9470 9475 9480 9485 9490 9495 9500 9505 9510 9515 9520 9525 9530 9535 9540 9545 9550 9555 9560 9565 9570 9575 9580 9585 9590 9595 9600 9605 9610 9615 9620 9625 9630 9635 9640 9645 9650 9655 9660 9665 9670 9675 9680 9685 9690 9695 9700 9705 9710 9715 9720 9725 9730 9735 9740 9745 9750 9755 9760 9765 9770 9775 9780 9785 9790 9795 9800 9805 9810 9815 9820 9825 9830 9835 9840 9845 9850 9855 9860 9865 9870 9875 9880 9885 9890 9895 9900 9905 9910 9915 9920 9925 9930 9935 9940 9945 9950 9955 9960 9965 9970 9975 9980 9985 9990 9995 9999 10000 10005 10010 10015 10020 10025 10030 10035 10040 10045 10050 10055 10060 10065 10070 10075 10080 10085 10090 10095 10099 10100 10101 10102 10103 10104 10105 10106 10107 10108 10109 10110 10111 10112 10113 10114 10115 10116 10117 10118 10119 10120 10121 10122 10123 10124 10125 10126 10127 10128 10129 10130 10131 10132 10133 10134 10135 10136 10137 10138 10139 10140 10141 10142 10143 10144 10145 10146 10147 10148 10149 10150 10151 10152 10153 10154 10155 10156 10157 10158 10159 10160 10161 10162 10163 10164 10165 10166 10167 10168 10169 10170 10171 10172 10173 10174 10175 10176 10177 10178 10179 10180 10181 10182 10183 10184 10185 10186 10187 10188 10189 10190 10191 10192 10193 10194 10195 10196 10197 10198 10199 10199 10200 10201 10202 10203 10204 10205 10206 10207 10208 10209 10210 10211 10212 10213 10214 10215 10216 10217 10218 10219 10220 10221 10222 10223 10224 10225 10226 10227 10228 10229 10230 10231 10232 10233 10234 10235 10236 10237 10238 10239 10240 10241 10242 10243 10244 10245 10246 10247 10248 10249 10250 10251 10252 10253 10254 10255 10256 10257 10258 10259 10260 10261 10262 10263 10264 10265 10266 10267 10268 10269 10270 10271 10272 10273 10274 10275 10276 10277 10278 10279 10280 10281 10282 10283 10284 10285 10286 10

5

- a. Assigns a globally unique CALS ID number to each independent CALS,
- b. Authenticates, maintains and distributes for replication by CALS a master database of all CALS-issued PubMbr/SrvcProvider IDs,
- c. Enforces interoperability among CALS so that a consumer end user of any PubMbr anywhere is technically capable of obtaining, information via HTTP request from any PubMbr globally, regardless of the PubMbr's CALS affiliation,
- d. Guarantees payments system wide,
- e. Collects interchange fees.

## 6.1 CROSS-SYSTEM AUTHENTICATION AND LOGGING -- FLOW DIAGRAM

10 The diagram submitted as Figure 2 provides a graphical depiction of the flow of authentication, content and logging among the various entities comprising TVS. The reader may refer to the diagram, which references the numbered paragraphs below. The Exhibit B legend contains the lexicon of abbreviations used in the numbered paragraphs.

15 6.1.1. A user member (CM) initiates a session request, typically by submitting a user name/password string via a common World Wide Web browser to the TVS http server daemon operated by the end-user's CSPa.

20 6.1.2. The TVS-enhanced server daemon executes its own custom process to verify that the CM seeking to begin a session has proper authorization to do so. The procedure may include a process to verify the CM's credit with an external source or database or to check the status of balance of an account against which purchases will be debited. If the CM is found to be properly authorized, CSPa's TVS-enhanced server daemon employs a pre-established UDP connection to its CALSa to (a) Submit user-profile information to CALSa for storage in the CALSa dynamic session database; and (b) Direct CALSa to issue and return via UDP a globally unique session token. This token is also stored in the CALSa dynamic session database and is referenced to the related user-profile information also contained there.

25

6.1.3. The CALS server returns the globally unique session token via UDP to the requesting CSPa. Its contents are encoded so they may be read only by CALSa and by other CSPs and

Ms within the CALSa service group. CSPa saves the token to assist with later reconciliation and billing of CM information purchases, if any.

6.1.4. CSPa responds to the CM's request to initiate a session by returning a copy of the session token to the CM's Web-browser software. Depending upon the TVS version implemented, the token is returned as (a) a query string appended following the syntax "?TVS=" to all URLs on the HTML page generated and returned to the Web browser; or, (b) as header information; or, (c) or by any other method standardized by TVS protocol adherents and/or licensees.

6.1.5. The CM user next now "clicks" on a URL which addresses a resource at CPMb, a publishing member which is not directly affiliated with the CALS of the CM's service provider. Submitted along with the URL as part of the HTTP request is a copy of the encoded session token.

6.1.6. In considering how to respond to the HTTP request, the TVS-enhanced Web-server daemon at CPMb strips off the session token (which it need not un-encode) and forwards it via a persistent UDP connection to its CALSb for analysis. Prior to returning a result or resource to the requesting CMa user member, it awaits a response.

6.1.7. CALSb attempts to un-encode the UDP-transmitted session token and, doing so successfully, finds, by referring to a database table of global CALS provided and updated by CIS, determines that it was generated by CALSa. Since the CALSb dynamic session database does not contain user-profile information for CMa, CALSb relays the token to CALSa for authentication via a UDP connection which it opens for this purpose. By this process, CALSa becomes financially responsible for the purchases of CMa globally. If CALSb is unable to un-encode the session token, it is discarded and CPMb is told that the user making the information request cannot be authenticated.

6.1.8. CALSa attempts to un-encode the UDP-transmitted session token received from CALSb and, doing so successfully, finds that it was generated by CALSa. Referring to its dynamic session database, CALSa finds the user CM's user profile, including class-of-service

and, depending upon the version of TVS deployed, credit/debit-limit and demographic/personalization data. CALSa employs the established UDP connection with CALSb to return the profile information.

6.1.9. CALSb relays the CMa profile information to CPMb. CPMb uses the profile information to make a series of decisions about how to respond the end user's information or access request. Among the decisions points which could lead to refusing the request:

- (a) After giving effect to the markup percentage set by CSPa and transmitted as part of the profile information, comparing the ultimate price to the end-using CMa of the resource requested and determining whether the price exceeds CMa's "query threshold."
- (b) Comparing the retail value of the information requested against the user's credit limit or debit-account balance, as received from CSPa.
- (c) Comparing the type of information requested against the user-defined flag to show or not show "parental-control" material.
- (d) Comparing the type of resource requested against the "permissions" granted to the service class of the CM making the request.

6.1.10. Assuming that CPMb finds no basis to invoke any rule against serving the original HTTP request, CPMb returns the requested resource via HTTP, along with a fresh copy of the original session token using the same transfer protocol employed in Step No. 4, above.

20 User has now "bought" information, if it is subject to a monetary charge.

6.1.11. Upon successful completing of the HTTP transmission to the end-user's Web browser, CPMb's TVS-enhanced logging daemon stores a record of the access in common log format enhanced with a number of TVS-specific fields (see Section XX for a description of the additional fields), and also transmit a copy of the access-log report to CALSb. The log report includes the PubMbr/SrvcProviderID of CSPa. Accordingly, CALSb relays the log

report to CALSa and takes no further action on it. Depending upon the version of TVS, the TVS-enhanced common log format data may include a field containing an XML-compliant Universal Document Identifier (UDI) extracted from the document or resource as well. The UDI should permit determination of the copyright ownership of the original document or resource, regardless of which CPM or CSP vended it. In this manner, TVS partners may "cache" each other's resources to improve network efficiency without affecting the ability of TVS to identify the correct recipient of the wholesale royalty at settlement time.

5           6.1.12. CALSa receive from CALSb a relayed copy of the access-log report generated by CPMb. Its logging daemon stores a copy in a database of all global access to TVS-enabled  
10           resources by users of CSPa for later settlement and billing.

15           6.1.13. Depending upon the version of TVS, CALSa also copies a log report to a real-time metering and billing utility which will permit: (a) The end-using CMa to request and review records of current session access by clicking to an address on a web server at CSPa. The request generates a call from CSPa to CALSa for current-session access logs for end-user CMa. The logs are then parsed against credit/debit account status, pricing and service-class rules maintained by CSPa for its end-users, and fed into a dynamically-generated page shown to the user; or, (b) The assembly and transmission by CSPa via Email to the end user once in each 24-hour cycle a compilation of all TVS-enabled resource purchases or accesses during the previous period from data provided on a batch basis from CALSa. This permits the end-  
20           user to verify and/or dispute charges shortly after they are incurred.

25           6.1.14. The "show the bill" processes are separate and distinct from the periodic settlement by each CALS of all royalty credits to member CPMs and all royalty-plus-transaction-fee debits to member CSPs. Settlement occurs in the prototype implementation across the bank Automated Clearing House network.

25           7. EXAMPLES OF USE

The diagrams in Figures 1 and 2 illustrate all the functional components of the TVS service.

### 7.1. CONTENT REQUEST + USER AUTHENTICATION

To begin, the user points his WWW browser to the home page set up for him at his "home" Publishing Member (step 1). This page has been designated as "authentication required" by the Publishing Member, so the user's browser receives back from the Publishing Member's HTTP server an appropriate status message. The browser prompts the user for his user-name and password, which it then returns to the HTTP server as Request Header information.

TVS does not affect the authentication model used by the HTTP server. Currently, most Web browsers use the Basic Authentication scheme which is not "secure". One additional scheme has been proposed to the IETF (so-called "Digest Access"), and several other schemes -- notably, the BellCore "one-time password" scheme -- are of interest. The intent of the new models is to provide a very high level of certainty in the authenticity of the user (client). Again, TVS does not dictate a specific model of "original authentication".

### 7.2 PROFILE "REGISTRATION" AND TOKEN REQUEST

Once the HTTP server has obtained the user's Authentication information and has validated it locally, the HTTP server contacts TVS with a request for a new Authentication Token. In making this request, the HTTP server sends the user's profile to TVS with a request for a new Authentication Token. This profile information (along with other per-user information) is stored in each publisher's registration database.

### 7.3 TOKEN GENERATION AND RETURN

TVS uses information from the user's profile to build the Authentication token. For example, the user's service class information is used to determine what the token's validity period will be. The Authentication Token has an encrypted "payload" and is "uuencoded" and "sanitized" to accommodate the Web URL naming syntax where required. The token is "opaque" to both the HTTP server and to the Web browser client.

TVS uses private-key encryption technology which is well-known to the Internet community and unencumbered by patent or export restrictions to the best of our knowledge.

#### 7.4 CONTENT AND AUTHENTICATION RETURN

5 When the HTTP server receives the returned token, it is ready to deliver the requested content (as well as the token) to the requesting client. The content is delivered in the canonical HTTP method (accompanied by MIME Response Headers as appropriate). The Authentication token can be delivered to the user's client program (Mosaic, Netscape, Lynx, an "agent", etc.) in several ways.

10 To accommodate current WWW browsers without modification, the Publishing Member's HTTP server "tags" all Clickshare URLs in the requested content with a query string that includes the token. However, this method is not ideal from a security standpoint, and presents difficulties if a rigid Content-Length needs to be enforced. A better model is to use HTTP 1.0 Request/Response Headers to transfer the token among parties "out of view". [See 5.18, above: "Other Modes for Transferring Token."]

15 Upon return of this original request, a TVS "session" has begun.

[Steps 7.1 through 7.4 occur only once per session.]

#### 7.5 CONTENT REQUEST WITH TOKEN

20 After receiving his original request (and having been returned a valid Authentication Token), the user will make subsequent requests for information. These requests can go to any number of services including the "home" Publishing Member, any other Publishing Member using TVS, or any other service not affiliated at all with Clickshare Service Corp or TVS. In the (we hope) common case where another TVS-enabled server is contacted, the user's browser will provide the Authentication Token to the server along with the information request.

## 7.6 USER VERIFICATION

The HTTP server contacts the TVS server to verify that the provided token is valid (that is, this is a valid user and a valid session).

## 7.7 VERIFICATION AND PROFILE RETURN

5 The TVS server receives the request, and verifies it using the internal databases it has constructed from the information provided when Authentication Tokens are issued. As an acknowledgment, TVS returns the user's profile information to the HTTP server.

## 7.8 CONTENT RETURN

10 The HTTP server uses the profile information to determine how best to respond to the user's request. In some cases, information in the profile may indicate that the server should not respond -- or warn the user about the cost or nature of the information requested. The profile information returned to the HTTP server can be used by the server itself to fulfill the request (typically the case with standard "static" file service requests), and is also made available as part of the execution environment for Common Gateway Interface (CGI) scripts.

15 **7.9 CONTENT ACCESS LOGGING**

After the HTTP server has returned the requested content to the user, this access is logged to the TVS service. A canonical log format is currently used, with information added in keyword=value form at the end of the record.

20 Steps 7.5 though 7.9 are repeated for every content/service request within a session when the user requests content from another TVS-enabled publisher. Requests sent to other (non-affiliated) HTTP servers are not effected.

## 7.10 HANDLING VALIDATION TIME-OUT

When a user's Authentication Token "times-out", information requests made with that token are invalid. If the user does not specifically end his session prior to this time-out, it is likely that the user will be making an information request to a Publishing Member other than his "home" when the time-out happens.

5        TVS, in cooperation with the HTTP servers, provides a mechanism to return the user to his "home" Publishing Member, undertake the process of re-authentication, and return to the site of the timed-out request - all transparently to the user. This process is handled using HTTP "Redirect" responses, but the key to success is the association with TVS which is the only party that knows where the user's home can be found.

10       A similar process works when completely invalid tokens are presented to TVS for verification. In such cases, TVS instructs the HTTP server to redirect the user to known points (in the current case, to Clickshare Service Corp.'s pages) such that the user can return "home" himself, or can select a "home" if necessary.

15       It is obvious that minor changes may be made in the form and construction of the invention without departing from the material spirit thereof. It is not, however, desired to confine the invention to the exact form herein shown and described, but it is desired to include all such as properly come within the scope claimed.

The invention having been thus described, what is claimed as new and desire to secure by Letters Patent is: